

# The Design of High Speed Wireless Video Monitoring System on Beagle Board Using ARM9 and GPRS

Golluri Venu

Asst.proff, ECE Dept, Christhu Jyothi Institute of Technology and Science,  
Jangaon, Warangal.

**Abstract—** In this paper, Using ARM9 as core with Ethernet or GPRS a remote video monitoring system is designed. The embedded chip and the programming techniques are adopted. First, USB camera video data are collected by the embedded Linux system, processed, compressed and transferred by the processing chip. Then, video data are sent to the monitor client by GRPS or Ethernet network. Tests show the presented wireless video surveillance system is reliable and stable. And it has a perfect application prospects with real-time monitor.

And in addition, the same is trying to implement on Beagle board which is approximately 150 times faster than the MINI2440 friendly ARM board and working out maintaining the database for storing such a huge or bulky live video data with the help of SQ LITE technique is possible.

**Keywords-** Embedded Linux; ARM core; Surveillance; GPRS

## 1. Introduction

Video surveillance systems play an increasingly important role to maintain social security. It has been widely used in many fields, such as finance, public security, banking, and home. Traditional video surveillance can generally achieve close distance monitoring, by using the PC as a monitor host, monitor host connected monitor camera with coaxial cable. With the development of IT technology, especially the rapid development in embedded CPU of the open Linux systems, video conferencing, remote monitoring, and mass data processing fields can apply the embedded technology. This embedded monitoring system to overcome the week points of the traditional video surveillance systems, such as complex structure, poor stability, and expensive cost. Meanwhile, it possess some advantages, for example, hardware and software can be cut, compact construct, portable, low power consumption, more suitable for long-distance transmission. In this system, we use the high-performance ARM9 chip as the microprocessor. Video data is captured from a USB camera, compressed into JPEG format, transferred the 3G network under the control of the ARM9 chip; then, the monitor client will receive the compressed data frame to restructure, and recompose video images. Wireless video monitor system provide a practical solution for remote wireless monitoring with low cost[1].

The entire system is consisting of the video capture, send and receive of the wireless network, to image data recover. The video capture is the core part of the system. The camera capture video data, the data compressed, and transferred into a Video stream format, this is pending transfer data. GPRS transceiver module is use for wireless network module [2][3]. Data can be transmitted into the frame format and sent to the been accepted, prepare it in two-column format, including 3G network; data arrival the receiving monitor through high- speed network transmission, this can also be a data center. Host to complete development system program development, debugging, etc.; the system structure is shown in Figure 1. The video signal monitor of the entire system is under the control of S3C2440

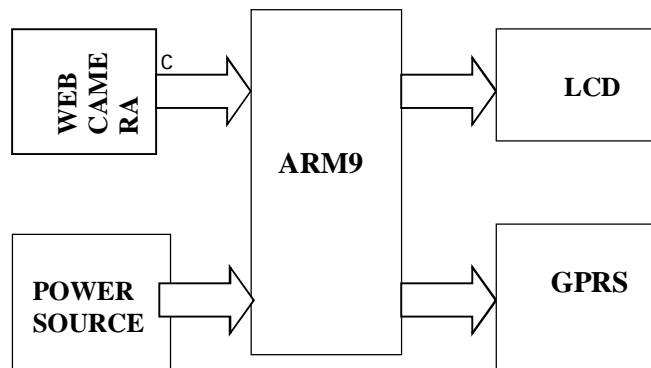


Fig 1. The working system

## II The System Design

### A. The System Hardware Design

The main processor uses the ARM9. The S3C2440 offers outstanding features with its CPU core, a16/32 bit ARM920T RISC processor designed by ARM, Ltd. The ARM920T implements MMU, AMBA BUS, and Harvard cache architecture with separate 16kB instruction and 16kB data caches, each with 8-word line length. It has 2-port USB Host and 1-port USB Device, support 64MB Flash and 64 MB SDRAM memory cell. GPRS module connected GPRS network through the RS32-expansion interface, the module with the address locator can also be extended, and system can take full advantage of the 3G network for data transmission. JTAG port is use for system debugging [4].

The design of GPRS Communication Module

In order to connect GPRS module with GPRS wireless communications network, the opened-GPRS-services external SIM card must be in use. SIM\_VCC is the mains to which the G20 uses to provide power, SIM\_Reset is used by G20 for the SIM card resetting control, SIM Clock is provided to the SIM card for clock, SIM\_DIO is for the data transmission connection between the G20 and the SIM card

### B. The System Software Design

#### 1. The Image Capturing

The video capture is the core of the entire video capture module. The video capture is finished by the V4L (the video4linux) of Embedded Linux operating system and imaging device drivers. V4L is a set of API of the Linux imaging systems and the Linux kernel support embedded images. The OV511 chip with built-in drive is used in the system[5][6]. Driver loading process is as follows: making menu config in the kernel directory, then selecting the Video for Linux usb support directory, and support for usb and usb camera the ov511 support, finally save and exit. The program can be loaded successfully. and the system starts with the new kernel after running the ov511. O

The system functions are as follows:

```

int v4l_open(char *dev, v4l_device *vd){ //Open device
intv4l_get_capability(v4l_device *vd){ //Read information
int v4l_get_picture(v4l_device* vd) { //Capturing images
    int v4l_close(v4l_device *vd){ //Close video device
    typedef struct v4l_struct
  
```

```
{
    int fd;

    struct video_capability capability; struct video_channel channel[4]; struct
    video_picture picture; struct video_window window; struct video_capture capture;
    struct video_buffer buffer;

    struct video_mmap mmap; struct video_mbuf mbuf; unsigned char
    *map;

    int frame;

    int framestat[2];

}vd;
```

The external device is managed by the device file in the Linux operating system. So, the operations of external devices are transformed into the operation of the device files. The main process is shown in Figure 3

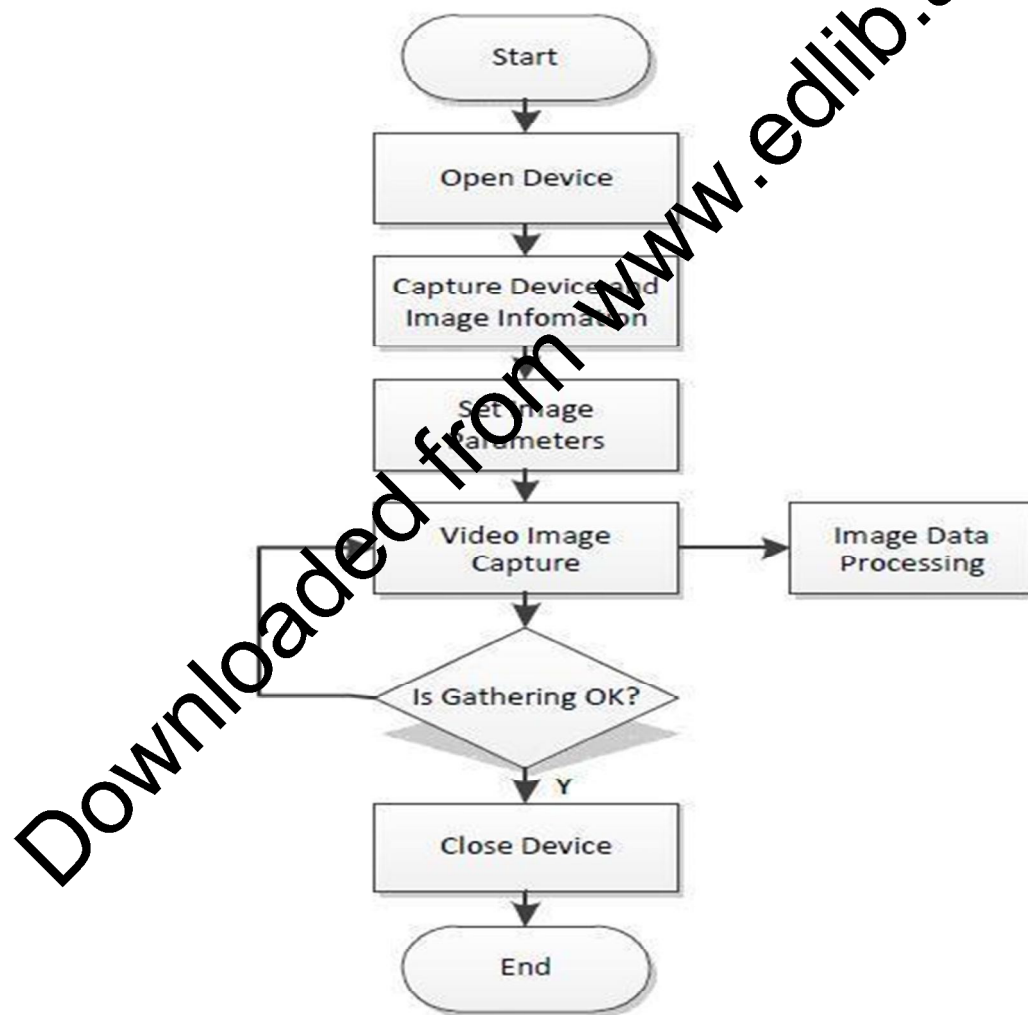


Fig 3. The image capturing process

After the above steps, the camera video data is acquired into memory. First the video data of collection is compressed into JPEG. Then the data is packaged into the data packet transmission to the video application server for processing.

The capture function is as follows: Loop:

```
{
    If(framestat[0] = 0) VIDIOCMCAPTURE(); //start
    if (grab_fd,VIDIOCSYNC,&(vd->frame_current)) framestat[1] = 1;
    if(framestat[1] = 1)
VIDIOCMCAPTURE();
if(grab_fd,VIDIOCSYNC,&(vd->frame_current)) framestat[0] =0; //wait until the complete
}
```

## 2. Video Compression

The JPEG compression encoding scheme is used in the system based on the DCT (Discrete Cosine Transform) sequence. The original image is divided into  $8 \times 8$  small video image after DCT transform. The low frequency components are concentrated in the upper left corner and high-frequency components are in the lower right corner (DCT transform is actually a low-pass filter of the spatial domain)[7].

The high spatial frequency for the human eye is far from low-frequency sensitive, so the treatment of visual loss is very small. The video image is clear and true by this method.

The data is quantified after DCT. The quantization is a value divided by the quantization table corresponding value. Due to the smaller value of the upper-left corner of the quantization table, the value of the upper right corner on the upper left corner of the low frequency components can be maintained, high-frequency component can be fully inhibited. After Quantified at high spatial frequencies a large numbers of consecutive zero will appear. The coding process is used. The system uses Huffman variable length coding to encode the quantized DCT coefficients, so that the entropy is minimum. The JPEG encoding under Linux can use the library Libjpeg, that is used under linux standard library, and its function is the picture in accordance with a certain percentage of compressed into JPEG format pictures or JPEG format to decompress. Many software operating on JPEG are based on this library.

## 3. The Program Design

System programming is divided into client and server programming. The client program realizes the data collection and sending and server program completes the data reception and decoding, display. The socket programming is used.

The socket is the interface of the same host between the application layer and transport layer. TCP or UDP may be used to provide concurrent services for more than one application process while communicating between application layer and transport layer[8][9]. In order to distinguish between different application processes and connections, many computer operating systems provide for the application by TCP / IP protocol. The interact is called the socket. Its main parameters are three: the destination IP address, transport layer protocol and port number. These three parameters can be distinguished from a different application process or a network connection communication, concurrent services of data transmission. The system uses the stream socket design whose communication process is shown in Figure 4.

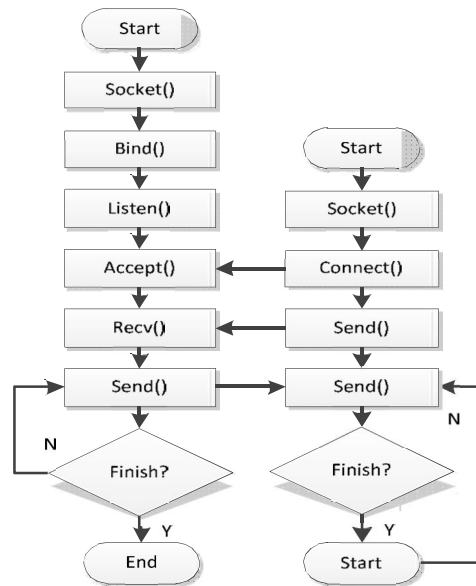


Fig 4. The TCP communication process between C/S

### I. Server Program Design

1. Create a socket: `sockkh ( AF_INET, SOCK- STREAM, 0);` // This function is used to create a new socket, AF\_INET for specified Internet address; SOCK-STREAM specify a stream socket; 0 means the default protocol.
2. Bind function: the `bind (sockkh, (struct sockaddr*) & The server_addr, sizeof (struct the sockaddr));` // Associated the physical location of the socket returned by socket port on the network. Sockkh parameters are socket descriptor returned; local address; the length of the socket address structure.
3. Listen (`sockkh, 20`); // Socket port accepts the connection request is sent from the client. The maximum number of clients is 20.
4. `Int accept (sockkh, struct sockaddr * addr, int * addrlen);` // receiving connection requests from CLIENT. addr is used to store client information. Filling Address and port when connected to the client; `addrlen` length numbers of client address bytes. `connect(client handle, (struct sockaddr*)&servadr, sizeof(struct sockaddr));` // Establish a connection

### II.Experimesntal Analysis

The system running is shown in Figure 5. The image size captured and compressed is 200 \* 150 and an average of 20kB in the system. The image is clearer and the monitoring point is reflected. With the information image-level changes of collection, data acquisition frame rate is around 5. The running is stable. After the compression size is adjusted, frame rate can be improved, but not smooth.

### III.Conclusions

The Design of High Speed Wireless Video Monitoring System on Beagle Board Using ARM9 and GPRS. in the paper. The system adopts the higher standard treatment of JPEG compression and storing live data using SQ LITE method on beagle board which is usually 150 times faster than the ARM MINI400 board. The test system is stable, reliable, clearer performance on video; to meet the monitoring requirements

under the difficult, complex environment the research provides a novel application scheme with the wide application prospect for the monitor of the remote environment.

### References

1. Zhong Chongquan. Research and application of remote monitoringbased on embedded web. Chinese Journal of Scientific Instrument, Vol1.27, pp.575-577, June.2006
2. The Design of Video Surveillance System Using S3C2440 and 3G Module International Journal of Engineering and Advanced Technology (IJEAT)ISSN: 2249 – 8958, Volume-3, Issue-1First Author Name: G.Venu Second Author Name: G.Anitha
3. Peng Daogang, Zhang Hao, Zhang Kai, Li Hui. Research and development of the remote I/O data acquisition system based on embedded ARM platform. 2009 International Conference on Electronic Computer Technology ICECT 2009,341-344
4. Dong Xie.ARM-based embedded remote monitoring and control system design Gateway. Modern electronic technology, 2006, pp. 85-88

Downloaded from [www.edlib.asdf.res.in](http://www.edlib.asdf.res.in)